

PRAKTIKUM PEMROGRAMAN FRAMEWORK

MODUL 4

Pengenalan Controller dan View pada Laravel



Disusun Oleh:
Purnama Anaking, S.Kom., M.Kom.

**PROGRAM STUDI S1 SISTEM INFORMASI
FAKULTAS TEKNOLOGI INFORMASI DAN BISNIS
INSTITUT TEKNOLOGI TELKOM SURABAYA
2023**

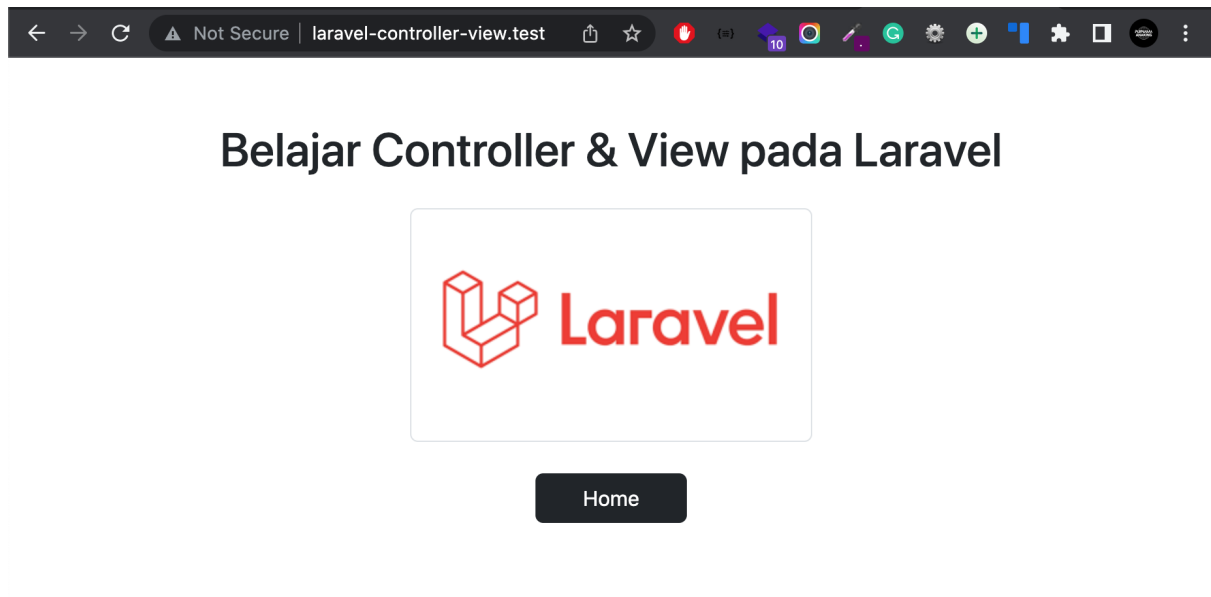
DAFTAR ISI

DAFTAR ISI	2
1. Generate Laravel Project dan Laravel UI	5
2. Bundling Asset dengan Vite	6
3. Install Bootstrap dan Bootstrap Icons Terbaru pada Project Laravel	8
4. Praktik Halaman Home	8
5. Praktik Halaman My Profile	10
6. Praktik Halaman Employee List	12
7. Praktik Halaman Create Employee	16
8. Tugas	19

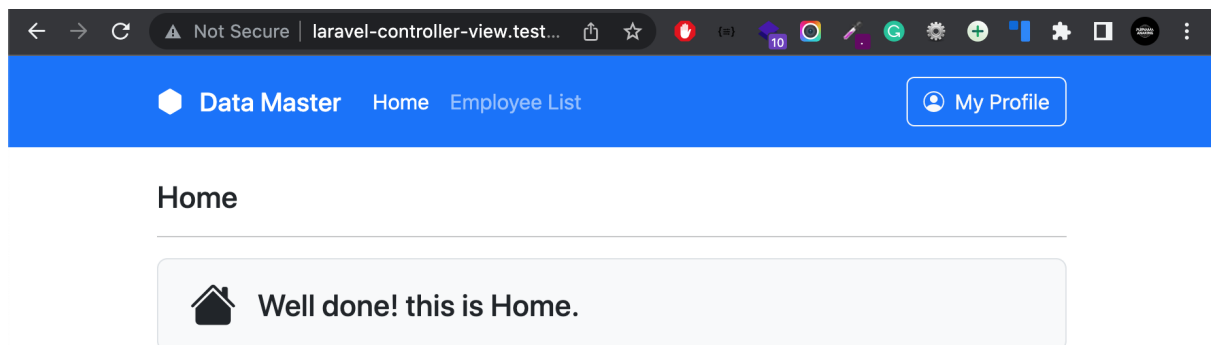
MODUL 4

PENGENALAN CONTROLLER DAN VIEW PADA LARAVEL

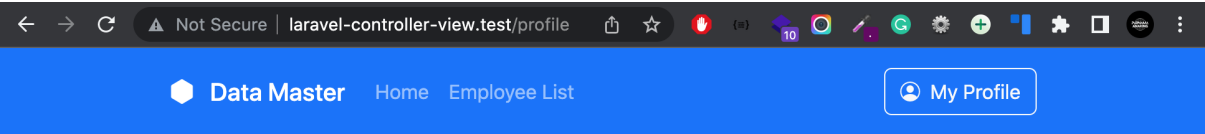
Pada praktikum kali ini kita akan melanjutkan belajar tentang controller dan view pada laravel. Materi tentang controller dan view akan diilustrasikan dalam bentuk studi kasus sederhana secara bertahap. Kegiatan ini agar mahasiswa dapat memahami dan menerapkan penggunaan controller dan view pada laravel framework.




Gambar. Hasil Akhir Bab 1, 2 dan 3



Gambar. Hasil Akhir Bab 4



Profile










 Well done! this is Profile.

Gambar. Hasil Akhir Bab 5



Employee List

Create Employee

First Name	Last Name	Email	Age	
Purnama	Anaking	purnama.anaking@gmail.com	20	  
Adzanil	Rachmadhi	adzanil.rachmadhi@gmail.com	25	  
Berlian	Rahmy	berlian.rahmy@gmail.com	23	  

Gambar. Hasil Akhir Bab 6

← → ↻ Not Secure | laravel-controller-view.test/employees/create

Data Master Home Employee List My Profile

Create Employee

First Name

Enter First Name

First name harus diisi.

Last Name

Enter Last Name

Last name harus diisi.

Email

Enter Email

Email harus diisi.

Age

Enter Age

Age harus diisi.

Cancel Save

Gambar. Hasil Akhir Bab 7

1. Generate Laravel Project dan Laravel UI

- Buat project laravel baru via composer

```
composer create-project laravel/laravel your-project-name
```

- Masuk ke dalam folder project laravel yang baru saja dibuat, kemudian install package Laravel UI.

```
composer require laravel/ui
```

- Generate scaffolding untuk project Laravel berbasis CSS framework Bootstrap.

```
php artisan ui bootstrap
```

- Jalankan script di bawah ini untuk compile scaffolding Bootstrap yang barusan di-setup.

```
npm install
```

- Jalankan script di bawah ini untuk mengaktifkan local development server Laravel sehingga aplikasi web dapat diakses melalui browser.

```
php artisan serve
```

2. Bundling Asset dengan Vite

- Download dan install NodeJS terbaru via <https://nodejs.org/en/download/>
- Periksa hasil instalasi Anda. Ketik script ini di prompt perintah Anda:

```
node -v
```

```
npm -v
```

- Install semua dependencies yang dibutuhkan untuk Bundling Asset dengan Vite, dengan mengetikkan perintah:

```
npm install
```

- Terapkan fitur “**Refreshing on Save**” dengan memeriksa file konfigurasi file vite pada **vite.config.js** dan sesuaikan seperti gambar di bawah ini.

```
import { defineConfig } from 'vite';
import laravel from 'laravel-vite-plugin';

export default defineConfig({
  plugins: [
    laravel({
      input: [
        'resources/sass/app.scss',
        'resources/js/app.js',
      ],
      refresh: true,
    }),
  ],
});
```

- Terapkan fitur “**Processing Static Assets With Vite**” dengan buka file **/resources/js/app.js** lalu sesuaikan kode program seperti di bawah ini. Vite akan merujuk pada path direktori yang kita definisikan untuk mengambil aset gambar/image yang kita butuhkan nantinya.

```
import './bootstrap';
import.meta.glob(['../images/**']);
```

- Buat folder pada **/resources** dengan nama **images**. Letakkan aset gambar/image yang akan kita gunakan pada website kita pada folder tersebut.
- Buka file View bernama **welcome.blade.php**. Hapus seluruh kode program yang ada. Kemudian, letakkan dan arahkan file **css** dan **javascript** yang telah didefinisikan di atas, pada file view (blade) menggunakan Blade Directive **@vite()**. Panggil asset gambar/image dengan pendekatan Vite seperti di bawah ini.

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Belajar Controller & View pada Laravel</title>
  @vite('resources/sass/app.scss')
</head>
<body>
  <div class="container text-center my-5">
    <h1 class="mb-4">Belajar Controller & View pada Laravel</h1>
    {{-- Contoh cara mereferensikan gambar di dalam file blade dengan
menggunakan pendekatan Vite --}}
    

    <div class="col-md-2 offset-md-5 mt-4">
      <div class="d-grid gap-2">
        <a class="btn btn-dark" href="">Home</a>
      </div>
    </div>
  </div>
  @vite('resources/js/app.js')
</body>
</html>

```

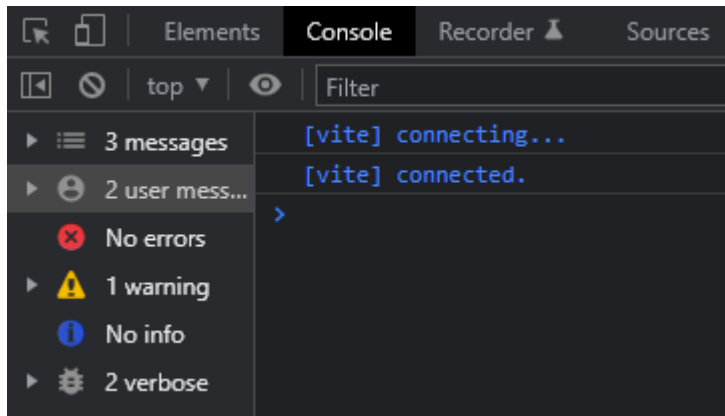
- Jalankan Vite.
 - Untuk mode **development** (jalankan ini saja jika sedang development):

```
npm run dev
```

- Atau untuk mode **production** (ketika akan deploy ke server):

```
npm run build
```

- Pastikan Vite Connected melalui Tab Console seperti di bawah ini.



- Jika sudah berhasil maka ketika kita mengubah kode program pada file css, blade, ataupun javascript, maka browser akan melakukan **reload secara otomatis**.

3. Install Bootstrap dan Bootstrap Icons Terbaru pada Project Laravel

- Jalankan perintah berikut pada project laravel anda untuk menginstall bootstrap, popper dan bootstrap icons terbaru.

```
npm install bootstrap@5.3.0-alpha3 bootstrap-icons @popperjs/core
```

- Di dalam project laravel anda, buka file **resources\sass\app.scss** dan tambahkan:

```
@import "bootstrap-icons/font/bootstrap-icons.css";
```

- Hapus atau Comment kode import Font & import Variables pada file **resources\sass\app.scss**. Kode program akan terlihat seperti di bawah ini:

```
// Fonts
// @import url("https://fonts.bunny.net/css?family=Nunito");

// Variables
// @import "variables";

// Bootstrap
@import "bootstrap/scss/bootstrap";
@import "bootstrap-icons/font/bootstrap-icons.css";
```

4. Praktik Halaman Home

- Generate sebuah **Basic Controller** dengan nama **HomeController** menggunakan perintah Artisan berikut ini. File Controller akan secara otomatis dibuatkan dan diletakkan pada folder **app/Http/Controllers/**.

```
php artisan make:controller HomeController
```

- Buka file **/routes/web.php** kemudian buat Route dengan kode seperti di bawah ini:


```
Route::get('home', [HomeController::class, 'index'])->name('home');
```

- Pastikan kode di bawah ini ada di bagian atas file **/routes/web.php**.

```
use App\Http\Controllers\HomeController;
```

- Sesuaikan kode program pada file **HomeController** tersebut seperti di bawah ini:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller
{
    function index()
    {
        $pageTitle = 'Home';

        return view('home', ['pageTitle' => $pageTitle]);
    }
}
```

- Buat file View dengan nama **home.blade.php** pada folder **resources/views/**.
Sesuaikan kode program pada file View tersebut seperti di bawah ini:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>{{ $pageTitle }}</title>
    @vite('resources/sass/app.scss')
</head>
<body>
    <nav class="navbar navbar-expand-md navbar-dark bg-primary">
        <div class="container">
            <a href="{{ route('home') }}" class="navbar-brand mb-0 h1"><i
class="bi-hexagon-fill me-2"></i> Data Master</a>

            <button type="button" class="navbar-toggler"
data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent">
                <span class="navbar-toggler-icon"></span>
            </button>

            <div class="collapse navbar-collapse">
```

```

id="navbarSupportedContent">
    <hr class="d-lg-none text-white-50">

    <ul class="navbar-nav flex-row flex-wrap">
        <li class="nav-item col-2 col-md-auto"><a href="{{
route('home') }}" class="nav-link active">Home</a></li>
        <li class="nav-item col-2 col-md-auto"><a href="{{
route('employees.index') }}" class="nav-link">Employee List</a></li>
    </ul>

    <hr class="d-lg-none text-white-50">

    <a href="{{ route('profile') }}" class="btn
btn-outline-light my-2 ms-md-auto"><i class="bi-person-circle me-1"></i>
My Profile</a>
    </div>
</div>
</nav>

<div class="container mt-4">
    <h4>{{ $pageTitle }}</h4>
    <hr>
    <div class="d-flex align-items-center py-2 px-4 bg-light
rounded-3 border">
        <div class="bi-house-fill me-3 fs-1"></div>
        <h4 class="mb-0">Well done! this is {{ $pageTitle }}.</h4>
    </div>
</div>

@vite('resources/js/app.js')
</body>
</html>

```

- Pertanyaan:
 - Jelaskan dan dokumentasikan setiap kode program yang anda kerjakan di atas!
 - Buka file View **welcome.blade.php**, arahkan button **Home** ke **Route** dengan nama **"home"** yang telah dibuat pada Bab ini.

5. Praktik Halaman My Profile

- Generate sebuah **Single Action Controller** dengan nama **ProfileController** menggunakan perintah Artisan berikut ini. File Controller akan secara otomatis dibuatkan dan diletakkan pada folder **app/Http/Controllers/**.

```
php artisan make:controller ProfileController --invokable
```

- Buka file **/routes/web.php** kemudian buat Route dengan kode seperti di bawah ini:

```
Route::get('profile', ProfileController::class)->name('profile');
```

- Pastikan kode di bawah ini ada di bagian atas file **/routes/web.php**.

```
use App\Http\Controllers\ProfileController;
```

- Sesuaikan kode program pada file **ProfileController** tersebut seperti di bawah ini:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ProfileController extends Controller
{
    /**
     * Handle the incoming request.
     */
    public function __invoke(Request $request)
    {
        $pageTitle = 'Profile';

        return view('profile', ['pageTitle' => $pageTitle]);
    }
}
```

- Buat file View dengan nama **profile.blade.php** pada folder **resources/views/**. Sesuaikan kode program pada file View tersebut seperti di bawah ini:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>{{ $pageTitle }}</title>
    @vite('resources/sass/app.scss')
</head>
<body>
    <nav class="navbar navbar-expand-md navbar-dark bg-primary">
        <div class="container">
            <a href="{{ route('home') }}" class="navbar-brand mb-0 h1"><i
class="bi-hexagon-fill me-2"></i> Data Master</a>

            <button type="button" class="navbar-toggler"
data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent">
                <span class="navbar-toggler-icon"></span>
            </button>
        </div>
    </nav>
</body>
</html>
```

```

        </button>

        <div class="collapse navbar-collapse"
id="navbarSupportedContent">
            <hr class="d-lg-none text-white-50">

            <ul class="navbar-nav flex-row flex-wrap">
                <li class="nav-item col-6 col-md-auto"><a href="{{
route('home') }}" class="nav-link">Home</a></li>
                <li class="nav-item col-6 col-md-auto"><a href="{{
route('employees.index') }}" class="nav-link">Employee List</a></li>
            </ul>

            <hr class="d-lg-none text-white-50">

            <a href="{{ route('profile') }}" class="btn
btn-outline-light my-2 ms-md-auto"><i class="bi-person-circle me-1"></i>
My Profile</a>
        </div>
    </div>
</nav>

<div class="container mt-4">
    <h4>{{ $pageTitle }}</h4>
    <hr>
    <div class="d-flex align-items-center py-2 px-4 bg-light
rounded-3 border">
        <div class="bi-person-circle me-3 fs-1"></div>
        <h4 class="mb-0">Well done! this is {{ $pageTitle }}.</h4>
    </div>
</div>

    @vite('resources/js/app.js')
</body>
</html>

```

- Pertanyaan:
 - Jelaskan dan dokumentasikan setiap kode program yang anda kerjakan di atas!

6. Praktik Halaman Employee List

- Generate sebuah **Resource Controller** dengan nama **EmployeeController** menggunakan perintah Artisan berikut ini. File Controller akan secara otomatis dibuatkan dan diletakkan pada folder **app/Http/Controllers/**.

```
php artisan make:controller EmployeeController --resource
```

- Buka file **/routes/web.php** kemudian buat Route dengan kode seperti di bawah ini:

```
Route::resource('employees', EmployeeController::class);
```

- Pastikan kode di bawah ini ada di bagian atas file **/routes/web.php**.

```
use App\Http\Controllers\EmployeeController;
```

- Sesuaikan kode program **function index()** pada file **EmployeeController** tersebut seperti di bawah ini:

```
public function index()
{
    $pageTitle = 'Employee List';

    return view('employee.index', ['pageTitle' => $pageTitle]);
}
```

- Buat folder baru dengan nama **/employee** pada **resources/views/**. Kemudian buat file View dengan nama **index.blade.php** di dalam folder tersebut. Sesuaikan kode program pada file View tersebut seperti di bawah ini:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>{{ $pageTitle }}</title>
    @vite('resources/sass/app.scss')
</head>
<body>
    <nav class="navbar navbar-expand-md navbar-dark bg-primary">
        <div class="container">
            <a href="{{ route('home') }}" class="navbar-brand mb-0 h1"><i
class="bi-hexagon-fill me-2"></i> Data Master</a>

            <button type="button" class="navbar-toggler"
data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent">
                <span class="navbar-toggler-icon"></span>
            </button>

            <div class="collapse navbar-collapse"
id="navbarSupportedContent">
                <hr class="d-lg-none text-white-50">

                <ul class="navbar-nav flex-row flex-wrap">
                    <li class="nav-item col-2 col-md-auto"><a href="{{
route('home') }}" class="nav-link">Home</a></li>
                    <li class="nav-item col-2 col-md-auto"><a href="{{
```

```

route('employees.index') }}" class="nav-link active">Employee
List</a></li>
    </ul>

    <hr class="d-lg-none text-white-50">

    <a href="{{ route('profile') }}" class="btn
btn-outline-light my-2 ms-md-auto"><i class="bi-person-circle me-1"></i>
My Profile</a>
    </div>
  </div>
</nav>

<div class="container mt-4">
  <div class="row mb-0">
    <div class="col-lg-9 col-xl-10">
      <h4 class="mb-3">{{ $pageTitle }}</h4>
    </div>
    <div class="col-lg-3 col-xl-2">
      <div class="d-grid gap-2">
        <a href="{{ route('employees.create') }}" class="btn
btn-primary">Create Employee</a>
      </div>
    </div>
  </div>
  <hr>
  <div class="table-responsive border p-3 rounded-3">
    <table class="table table-bordered table-hover table-striped
mb-0 bg-white">
      <thead>
        <tr>
          <th>First Name</th>
          <th>Last Name</th>
          <th>Email</th>
          <th>Age</th>
          <th></th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>Purnama</td>
          <td>Anaking</td>
          <td>purnama.anaking@gmail.com</td>
          <td>20</td>
          <td>
            <div class="d-flex">
              <a href="{{ route('employees.show',
['employee' => 1]) }}" class="btn btn-outline-dark btn-sm me-2"><i
class="bi-person-lines-fill"></i></a>
              <a href="{{ route('employees.edit',
['employee' => 1]) }}" class="btn btn-outline-dark btn-sm me-2"><i

```

```

class="bi-pencil-square"></i></a>

        <div>
            <form action="{{
route('employees.destroy', ['employee' => 1]) }}" method="POST">
                @csrf
                @method('delete')
                <button type="submit" class="btn
btn-outline-dark btn-sm me-2"><i class="bi-trash"></i></button>
            </form>
        </div>
    </div>
</td>
</tr>
<tr>
    <td>Adzanil</td>
    <td>Rachmadhi</td>
    <td>adzanil.rachmadhi@gmail.com</td>
    <td>25</td>
    <td>
        <div class="d-flex">
            <a href="{{ route('employees.show',
['employee' => 2]) }}" class="btn btn-outline-dark btn-sm me-2"><i
class="bi-person-lines-fill"></i></a>
            <a href="{{ route('employees.edit',
['employee' => 2]) }}" class="btn btn-outline-dark btn-sm me-2"><i
class="bi-pencil-square"></i></a>
        </div>
    </td>
</tr>
<tr>
    <td>Berlian</td>
    <td>Rahmy</td>
    <td>berlian.rahmy@gmail.com</td>
    <td>23</td>
    <td>
        <div class="d-flex">
            <a href="{{ route('employees.show',
['employee' => 3]) }}" class="btn btn-outline-dark btn-sm me-2"><i
class="bi-person-lines-fill"></i></a>
            <a href="{{ route('employees.edit',

```

```

['employee' => 3]) }}" class="btn btn-outline-dark btn-sm me-2"><i
class="bi-pencil-square"></i></a>

        <div>
            <form action="{{
route('employees.destroy', ['employee' => 3]) }}" method="POST">
                @csrf
                @method('delete')
                <button type="submit" class="btn
btn-outline-dark btn-sm me-2"><i class="bi-trash"></i></button>
            </form>
        </div>
    </div>
</td>
</tr>
</tbody>
</table>
</div>
</div>

@vite('resources/js/app.js')
</body>
</html>

```

- Pertanyaan:
 - Jelaskan dan dokumentasikan setiap kode program yang anda kerjakan di atas!
 - Tunjukkan dan buktikan ada Route apa saja yang dihasilkan pada Bab ini!

7. Praktik Halaman Create Employee

- Sesuaikan kode program **function create()** pada file **EmployeeController** tersebut seperti di bawah ini:

```

public function create()
{
    $pageTitle = 'Create Employee';

    return view('employee.create', compact('pageTitle'));
}

```

- Buat file View dengan nama **create.blade.php** pada **/resources/views/employee/**. Sesuaikan kode program pada file View tersebut seperti di bawah ini:

```

<!DOCTYPE html>
<html lang="en">
<head>

```



```

<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<title>{{ $pageTitle }}</title>
@vite('resources/sass/app.scss')
</head>
<body>
    <nav class="navbar navbar-expand-md navbar-dark bg-primary">
        <div class="container">
            <a href="{{ route('home') }}" class="navbar-brand mb-0 h1"><i
class="bi-hexagon-fill me-2"></i> Data Master</a>

            <button type="button" class="navbar-toggler"
data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent">
                <span class="navbar-toggler-icon"></span>
            </button>

            <div class="collapse navbar-collapse"
id="navbarSupportedContent">
                <hr class="d-lg-none text-white-50">

                <ul class="navbar-nav flex-row flex-wrap">
                    <li class="nav-item col-2 col-md-auto"><a href="{{
route('home') }}" class="nav-link">Home</a></li>
                    <li class="nav-item col-2 col-md-auto"><a href="{{
route('employees.index') }}" class="nav-link">Employee List</a></li>
                </ul>

                <hr class="d-lg-none text-white-50">

                <a href="{{ route('profile') }}" class="btn
btn-outline-light my-2 ms-md-auto"><i class="bi-person-circle me-1"></i>
My Profile</a>
            </div>
        </div>
    </nav>

    <div class="container-sm mt-5">
        <form action="{{ route('employees.store') }}" method="POST">
            @csrf
            <div class="row justify-content-center">
                <div class="p-5 bg-light rounded-3 border col-xl-6">

                    @if ($errors->any())
                        @foreach ($errors->all() as $error)
                            <div class="alert alert-danger
alert-dismissible fade show">
                                {{ $error }}
                                <button type="button" class="btn-close"
data-bs-dismiss="alert" aria-label="Close"></button>

```



```
@vite('resources/js/app.js')
</body>
</html>
```

- Sesuaikan kode program **function store()** pada file **EmployeeController** tersebut seperti di bawah ini:

```
public function store(Request $request)
{
    $messages = [
        'required' => ':Attribute harus diisi.',
        'email' => 'Isi :attribute dengan format yang benar',
        'numeric' => 'Isi :attribute dengan angka'
    ];

    $validator = Validator::make($request->all(), [
        'firstName' => 'required',
        'lastName' => 'required',
        'email' => 'required|email',
        'age' => 'required|numeric',
    ], $messages);

    if ($validator->fails()) {
        return redirect()->back()->withErrors($validator)->withInput();
    }

    return $request->all();
}
```

- Pertanyaan:
 - Jelaskan dan dokumentasikan setiap kode program yang anda kerjakan di atas!
 - Tunjukkan penerapan “**Passing Data to Views**” pada Bab ini.
 - Tunjukkan penerapan “**CSRF Protection**” pada Bab ini.
 - Terapkan “**Displaying The Validation Errors**” pada Bab ini yang hasil akhirnya seperti gambar di bawah ini.

The screenshot shows a web browser window with the address bar displaying 'laravel-controller-view.test/employees/create'. The page has a blue header with 'Data Master', 'Home', 'Employee List', and a 'My Profile' button. The main content area is a light gray box titled 'Create Employee' with a user icon. It contains four input fields arranged in a 2x2 grid. Each field has a placeholder text and a red error message below it. The fields are: First Name (placeholder: 'Enter First Name', error: 'First name harus diisi.'), Last Name (placeholder: 'Enter Last Name', error: 'Last name harus diisi.'), Email (placeholder: 'Enter Email', error: 'Email harus diisi.'), and Age (placeholder: 'Enter Age', error: 'Age harus diisi.'). At the bottom of the form are two buttons: 'Cancel' and 'Save'.

- Terapkan “**Repopulating Forms**” pada Bab ini sehingga ketika muncul pesan error pada saat submit form, nilai yang sudah diisi pada form tidak ter-reset tetapi tetap muncul pada form tersebut.

8. Tugas

1. **Praktekan** seluruh poin praktikum dan **jawab** seluruh poin pertanyaan yang ada di atas secara **INDIVIDU**.
2. Dokumentasikan seluruhnya (**screenshot kode program, output pada browser, jawaban pertanyaan**) dalam bentuk Laporan Praktikum.
3. **Upload manual** project laravel hasil praktikum ke github (hapus folder **node_modules & vendor**).
 - a. Buat **repository public** baru pada akun github anda.
 - b. Klik link “**Uploading an existing file**” untuk meng-upload project laravel yang telah anda buat.
 - a. Cantumkan **URL repository** yang telah dibuat pada **Laporan Praktikum**.
4. Kumpulkan Laporan Praktikum (**.pdf**) yang telah dibuat di dalam praktikum via **E-Learning** paling lambat sebelum jadwal praktikum minggu depan.